

Requirements engineering tools: Capabilities, survey and assessment

Juan M. Carrillo de Gea^{a,*}, Joaquín Nicolás^a, José L. Fernández Alemán^a, Ambrosio Toval^a,
Christof Ebert^b, Aurora Vizcaíno^c

^aSoftware Engineering Research Group, University of Murcia, Murcia, Spain

^bVector Consulting Services, Stuttgart, Germany

^cAlarcos Research Group, University of Castilla-La Mancha, Ciudad Real, Spain

ARTICLE INFO

Article history:

Received 16 September 2011

Received in revised form 20 March 2012

Accepted 30 April 2012

Available online 7 May 2012

Keywords:

Requirements engineering tools

Survey

ISO/IEC TR 24766:2009

ABSTRACT

Context: There is a significant number of requirements engineering (RE) tools with different features and prices. However, existing RE tool lists do not provide detailed information about the features of the tools that they catalogue. It would therefore be interesting for both practitioners and tool developers to be aware of the state-of-the-art as regards RE tools.

Objective: This paper presents the results of a survey answered by RE tool vendors. The purpose of the survey was to gain an insight into how current RE tools support the RE process by means of concrete capabilities, and to what degree.

Method: The ISO/IEC TR 24766:2009 is a framework for assessing RE tools' capabilities. A 146-item questionnaire based principally on the features covered by this international guideline was sent to major tool vendors worldwide. A descriptive statistical study was then carried out to provide comparability, and bivariate correlation tests were also applied to measure the association between different variables. A sample of the tools was subjected to neutral assessment and an interrater reliability analysis was performed to ensure the reliability of the results.

Results: The 38 participants sent back their answers. Most tools are delivered under a proprietary license, and their licenses are not free. A growing number of them facilitate Web access. Moreover, requirements elicitation exemplifies the best supported category of features in this study, whereas requirements modeling and management are the most badly supported categories.

Conclusion: The RE process seems to be well covered by current RE tools, but there is still a certain margin for amelioration, principally with regard to requirements modeling, open data model and data integration features. These subjects represent areas for improvement for RE tool developers. Practitioners might also obtain useful ideas from the study to be taken into account when selecting an appropriate RE tool to be successfully applied to their work.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

The activities in requirements engineering (RE) are highly dependent on human decisions and thus are difficult to automate [1]. However, software analysts are becoming more and more conscious of the benefits that can be obtained from automated support for RE. There is an important, increasing number of RE tools currently available on the market: a total of 100 RE tools can be found throughout RE resources on the Internet which host RE tool lists. These tools might not adequately support the activities of the RE process in the absence of certain features, and a risk thus exists in that current RE tools do not meet analysts' expectations. Since

the requirements for complex systems are themselves complex information that must be handled in complex processes, there are many strong requirements concerning a tool for managing them [2]. It would therefore be interesting to gain insights into what these desirable features are, and how they are supported by RE tools, given that existing RE tool lists do not provide detailed information about the tools they catalogue.

The main objective of this paper is to shed light on the state-of-the-art as regards RE tools' capabilities. Furthermore, experimentation in software engineering is necessary in order to achieve credible knowledge [3]. In this paper we therefore apply some well-known descriptive statistical analysis techniques to provide a rigorous overview of current RE tools. To this end, a 146-point framework has first been defined so as to encompass important features that should be supported by an RE tool. This framework is primarily made up of features drawn from the ISO/IEC TR 24766:2009 – “Guide for requirements engineering tool capabilities”, a new guide that

* Corresponding author.

E-mail addresses: jmcdg1@um.es (J.M. Carrillo de Gea), jnr@um.es (J. Nicolás), aleman@um.es (J.L. Fernández Alemán), atoval@um.es (A. Toval), christof.ebert@vector.com (C. Ebert), aurora.vizcaino@uclm.es (A. Vizcaíno).

recommends the features that should characterize an RE tool. A survey has then been carried out among 100 RE tool vendors in order to discover how their tools support the features of this framework. The complete list of RE tools was obtained from a set of relevant databases. The survey contains a specific part devoted to collaborative work and global software development (GSD) capabilities, but this is not described in this manuscript owing to space constraints. Finally, the responses from the 38 respondents have been used to depict the current situation as regards RE tools.

This manuscript is an assessment of the current RE tool marketplace through a detailed description of its strengths and weaknesses. A study of the general RE capabilities of concrete RE tools characterising the final scores of the participants can be consulted for more information [4].

The remainder of this paper is organized as follows: Section 2 provides comments on the related work. Section 3 presents a framework concerning the features that RE tools should provide. Section 4 shows the research methodology followed. The results of the study are then summarized and discussed in Section 5. Our conclusions and future work are highlighted in Section 6. Finally, Appendix A shows details of the classification framework.

2. Related work

This section shows related work concerning RE tools, RE tool comparison frameworks and RE tool surveys.

2.1. RE tools

There is a number of papers describing the development and capabilities of those RE tools that are suitable for specific or general purposes. Their associate RE method is normally also depicted. Some of the most recent are summarised as follows. Gregoriades and Sutcliffe [5] described a method and a tool called SRA with which to validate non-functional requirements in complex socio-technical systems. This tool can validate system reliability and operational performance requirements using scenario-based testing. Hall [6] presented the motivations for and problems with large scale scenarios, and a method called LSS, which uses automated and semi-automated techniques for description, maintenance and communication, with the use of large scale scenarios in RE. Two application domains are used to illustrate the approach: live military training instrumentation and electronic mail servers. Kääriäinen et al. [7] reported an unsuccessful experience with the development of an RE tool for extreme programming (XP): Story-manager. These authors affirm that XP requires an extensive set of tools to enable an effective execution of its practices. Giorgini et al. [8] proposed a tool called ST-Tool for the design and verification of functional and security requirements, as security must be dealt with early on during the requirements phase. Sey et al. [9] described the potential use of mobile tools for requirements engineers. In this context, Maiden et al. [10] reported empirical research to explore the use of mobile RE tools in practice. A mobile scenario tool developed by the authors to discover requirements directly in the user's work context is described. The results demonstrate that these tools can support the discovery and documentation of workplace requirements. Jiang and Eberlein [11] presented an RE tool that relies on knowledge to support RE process development and the selection of RE techniques. Unlike other tools, it uses knowledge representation to manage the knowledge of the RE process and its technique, thus assisting in the development of the most suitable RE process for a software project. Other studies [12,13] presented software development tools which can be synchronized with RE tools, for example for requirements collaborative access or for administrative purposes.

Some author's research work is based on extending the functionality of an existing RE tool. For instance, Schmid et al. [14] claimed that even though RE tools are widespread, the range of professional tool support for product line development is very poor. Thus, the authors identified the requirements that a tool extension for product lines must support, based on product line concepts and the functionality that existing RE tools support. In addition, these authors present an extension based on DOORS called REMAP-tool. Toval et al. [15] also identified eight key issues to be considered for an effective and practical reuse-based RE process, and developed an extension for RequisitePro called SirenTool in order to achieve requirements reuse.

2.2. RE tool comparison frameworks

Given that selecting an RE tool is not straightforward, it is necessary to follow certain guidelines to perform such a task. Homann et al. [2] reported a framework with which to help users and tool providers compare and select RE tools, and direct future tool developments, respectively. The authors presented a requirements catalogue for RE tools based on substantial project experience in the area of automotive, aircraft and defense systems. Gotel and Mäder [16] provided high-level guidance on designing an RE solution and selecting an RE tool. The minimum and desirable requirements, based upon the type and size of project, were examined, and the question of whether an RE tool is needed at all is also considered. Hamann and Oort [17] summarised the requirements and verification management process at a medium-sized space industry. They then characterised the requirements of the tool support needed, and provided both a functional (general, requirements management and verification control features) and a cost and effort evaluation of both custom-made and commercial tools. Hong et al. [18] discussed the characteristics to be provided in RE tools which can support management from an initial phase to the customer acceptance phase. A new tool for requirements management with explicit support for traceability evolution was also developed. The proposed traceability links and the ways in which to evolve them were described. Johansson and de Carvalho [19] investigated general RE tools and concluded that existing RE tools do not properly support requirements management in the enterprise resource planning (ERP) systems context. Practical guidelines were then provided on how to develop an RE tool that can be used when developing future ERPs. Heindl et al. [20] studied the selection of RE tools in large software and systems engineering companies, where several RE tools are usually in use. The authors reported on a value-based RE tool selection approach that helps to discover the optimal tool support, based on rating the value contribution of suitably-defined tool features for the given project context. Alenljung and Persson [21] proposed a summative, criteria-based evaluation method called DESCRY, which is empirically and theoretically grounded, in order to investigate to what extent RE tools have decision-supporting capabilities that improve decision support for RE decision-makers.

2.3. RE tool surveys

Surveys conducted on RE tools are neither numerous nor up-to-date. In addition, they are usually limited to a specific RE activity, and do not encompass general RE features. Moreover, existing surveys on RE tools are not supported by a formal guideline with which to evaluate RE tools. Schmid et al. [22] presented a survey on the simulation and animation capabilities of ten RE tools, with the purpose of validating requirements specifications—i.e. checking whether requirements specifications meet stakeholders' expectations—. Zowghi and Coulin [23] provided a survey on the important aspects of the techniques and approaches, in addition

to an overview of general and specific tool support for requirements elicitation. The authors examined the issues, trends, and challenges confronted by researchers and practitioners during the process of seeking, uncovering, acquiring, and elaborating requirements for computer based systems. Beuche et al. [24] suggested important requirements that must be observed if RE tools are to be usefully applied to product lines. Four notorious RE tools were evaluated on the basis of these requirements in their daily industrial use. The requirements presented indicate the future direction of tool development and method research. Toval et al. [15] analysed three popular, contemporary RE tools with the aim of checking their reuse capabilities in the light of a comparative framework, revealing a lack of automated support. The framework included: (i) the eight key issues identified for a practical requirements reuse; and (ii) the general needs of an RE tool according to a survey on RE tools carried out by the INCOSE. Carrillo de Gea et al. [4] surveyed 37 RE tool vendors and compared the results with user experiences from typical RE use cases.

This paper presents a detailed description of which is, to the best of our knowledge, the first industrial survey on the support that RE tools provide in the entire RE process. Moreover, we are not aware of any other study that takes into account a formal guideline for identifying relevant RE tool features. Our main purpose is to provide requirements engineers with an updated overview on the state-of-the-art of RE tools by highlighting their capabilities and potentials.

3. Classification framework

Gotel and Mäder [16] state that procuring an RE tool can be one of the most costly decision an organization can make. They also note the existence of a number of evaluation frameworks which might lead RE consultants and researchers to assist practitioners in selecting RE tools. Moreover, standards have already been shown to be of great use in supporting different work proposals, owing to the official and agreed background they provide. For instance, the ISO/IEC 15408 – “Evaluation criteria for IT security” standard has been successfully applied to the specification of security quality requirements for security critical systems [25]. Thus, before creating the survey we looked for an extensive framework to study RE tools, and we finally adopted the ISO/IEC TR 24766 [26] as the basis of a classification framework for the evaluation of relevant RE tool features, since to the best of our knowledge, it is the only formal guideline which proposes the set of capabilities that an RE tool should support.

The ISO/IEC TR 24766 is not, properly speaking, a standard, but a *Technical Report (TR) of Type 2*, meaning that there is a future but not immediate possibility of an agreement on an International Standard. It supplements ISO/IEC 14102:2008 – “Guideline for the evaluation and selection of CASE tools” [27], a standard which is focused on a more general concern: Computer-Aided Software Engineering (CASE) tools. The ISO/IEC TR 24766 classifies the RE tool capabilities into six major categories: *requirements elicitation*, *requirements analysis*, *requirements specification*, *requirements verification and validation*, *requirements management* and *other capabilities*. Each category encompasses a wide range of features conceived to support the corresponding activity within the entire RE process, with the exception of the *other capabilities* category, which includes features not linked to any other category of features.

Although the definition of our classification framework is strongly based on the ISO/IEC TR 24766, we deemed it appropriate to modify its original set of categories of features with the purpose of refining and rearranging some capabilities into two new and relevant categories, *requirements modeling* and *requirements traceability*: (a) on the one hand, the lack of modeling features in the ISO/

IEC TR 24766, despite the importance of such concerns, led us to complete our classification framework with new capabilities in requirements modeling; (b) on the other hand, there are a lot of features regarding traceability in the ISO/IEC TR 24766, but they are fairly spread out throughout the whole document and sometimes seem to be repeated. In our opinion the newly defined categories of features provide our classification framework with better order and understandability. Hence, the RE tools’ capabilities in our classification framework belong to one of the categories of features depicted below:

1. **Elicitation.** This category includes features focused on the ability of the tools to support the identification of stakeholders, the capturing and tracing of business/user requirements, functional requirements, and non-functional requirements during elicitation work.
2. **Analysis.** This category includes capabilities aimed at decomposing high-level requirements into details, evaluating feasibility, negotiating priorities, identifying conflicts, determining unclear, incomplete, ambiguous or contradictory requirements and resolving all these issues.
3. **Specification.** This category encompasses features focused on documenting the functions that a software or system must provide, and the constraints that it must respect, specified in a consistent, accessible and reviewable manner, in order to be able to accomplish this goal.
4. **Modeling.** This category includes features focused on the application of certain techniques to produce useful and verifiable requirements models.
5. **Verification and validation (V&V).** This category includes capabilities aimed at supporting the various tests and means of evaluation used in verifying and validating the requirements.
6. **Management.** This category explores the ability of the tools to support the monitoring of the changes and the maintenance of the requirements, thus ensuring that the requirements accurately reflect the product.
7. **Traceability.** This category includes capabilities focused on documenting the life of a requirement, providing linkage mechanisms between associated requirements, and tracking changes made to each requirement.
8. **Other capabilities.** This category encompasses features related to the integration of the tool into the system and software development environment.

A comparison between the number of capabilities falling within each category of features in the ISO/IEC TR 24766 and in our classification framework is provided in Table 1. 157 features were initially obtained from the ISO/IEC TR 24766. However, the full set of items gathered in this document seemed too complex to be adopted without modification. We thus adapted and/or aggregated these aforementioned capabilities and obtained a reduction of

Table 1
Number of capabilities.

Category (TR)	No.	Category (framework)	No.
Elicitation	37	Elicitation	20
Analysis	36	Analysis	11
Specification	16	Specification	7
		Modeling	13
Verification and validation	34	Verification and validation	11
Management	17	Management	18
		Traceability	29
Other capabilities	17	Other capabilities	17
Total	157	Total	126

Table 2
Goal/question metric template.

Goal	The goal is to depict the state-of-the-art of RE tools
Question	Do current RE tools adequately support the RE process?
Metric	The capabilities supported by the RE tools within distinct categories of features in a 146-item, ISO TR 24766-based classification framework
<i>Goal definition template</i>	
Object of study	The objects studied are the RE tools
Purpose	The purpose is to characterize the relevant RE tools' features and to evaluate the scene of RE tools
Quality focus	The quality focus is the effectiveness, cost and presence of the RE tools
Perspective	The perspective is from the point of view of the researcher and customer of RE tools
Context	The study is run using RE tools' vendors as subjects based on a questionnaire, and RE tools appearing in at least one of the databases described in Table 3 as objects

almost 20% in the amount of features, cutting them down from 157 to 126, even after including several additional capabilities inspired by our experience in industry. Both the newly defined categories of features and the new features themselves allowed us to better tailor our classification framework to our needs.

The reader is encouraged to review Appendix A for further discussion on the classification framework, since it provides an expanded explanation of its details and critical aspects, including the subdivision of the aforementioned categories of features into more specific groups of features.

4. Research methodology

This section provides detailed information on the procedure followed to design and conduct the study in which the state-of-the-art of RE tools was investigated by means of a survey.

4.1. Research goals

The research goals were outlined by using the Goal/Question Metric (GQM) paradigm [28], a purposeful approach with which to obtain a specification of a measurement model [29], which allowed us to: (1) focus on the formulation of the main goal of the research; (2) develop the question based on this goal; and (3) associate the question with the appropriate metric. The GQM approach can be used to elicit metrics that support decision processes in organizations [30], and it was thus appropriately applied to our study, which is focused on supporting the selection of an RE tool. The GQM goal template can be used to articulate the purpose of any study [3]. The GQM template used in this study is shown in Table 2.

4.2. Hypotheses

Starting from the categories of the RE tool classification framework described in Section 3, the following hypotheses are proposed in this study to depict the state-of-the-art of RE tools:

- H1.** Current RE tools adequately support:
- H1.1.** Requirements elicitation.
 - H1.2.** Requirements analysis.
 - H1.3.** Requirements specification.
 - H1.4.** Requirements modeling.
 - H1.5.** Requirements V&V.
 - H1.6.** Requirements management.
 - H1.7.** Requirements traceability.
 - H1.8.** Other capabilities.
 - H1.9.** All the above features.

The following variables are considered in this study in order to verify the hypotheses, one for each category of features: *elicitation, analysis, specification, modeling, V&V, management, traceability and other capabilities*. The value of a concrete variable is the number of capabilities within the associated category of features which are supported by the tools, divided by the total number of capabilities in that category (i.e. the number of affirmative answers given by each tool vendor for the category in question divided by the number of questions). Moreover, there is a *global* variable representing the tools' support of the previously stated categories all together, i.e. when these categories are considered in unison.

There are two other variables in this study: a variable describing the *cost per individual license*, and a variable representing the number of *licenses in use*, which were formerly qualitative ordinal variables but have been converted into quantitative variables in order to perform correlation tests. The total number of variables in this study is therefore 11.

4.3. Instrumentation

This study used four instruments: (a) RE tools' databases for the selection process; (b) an analysis method to evaluate RE tools; (c) a survey application; and (d) a questionnaire to collect the information on the RE tools selected.

The problem of selecting the set of tools to be included in the study was tackled by consulting seven databases which contain a series of lists of RE tools: *Ian Alexander, Alarcos Research Group, INCOSE, Ludwig Consulting Services, Qaguild, Volere, and @WEBO*. At the moment of accessing these databases, from July–August 2010, the number of tools specified in Table 3 was retrieved. A large number of tools appeared in more than one database simultaneously. In addition, a number of the tools listed were not in force, there was no vendor or person responsible for them, or their name had changed. After reviewing the whole set of obtained tools, a total amount of 100 RE tools was therefore chosen for evaluation; in the meantime, their vendors' contact details were collected.

The analysis method chosen was DESMET [31], which is a method designed by Kitchenham to evaluate software engineering methods and tools. DESMET defines distinct quantitative and qualitative evaluation types. Moreover, it uses the term *feature analysis* to describe a qualitative evaluation. Feature analysis is based on identifying the requirements that users have for a particular task or activity and mapping them onto the features that a tool aimed at supporting that task or activity should possess. An evaluator then assesses the level of provision of the identified features by a number of alternative tools. Feature analysis is referred to as qual-

Table 3

Databases hosting RE tools lists. A: Ian Alexander, G: Alarcos Research Group, I: INCOSE, L: Ludwig Consulting Services, Q: Qaguild, V: Volere, W: @WEBO.

Database	Web	No.
A	http://easyweb.easynet.co.uk/iany/other/vendors.htm	67
G	http://sites.google.com/site/toolsgsd/tools-1/software-requirement-tools	7
I	http://www.incose.org/ProductsPubs/products/rmsurvey.aspx	34
L	http://www.jiludwig.com/Requirements_Management_Tools.html	40
Q	http://qaguild.com/Toolsdirectory/RequirementManagementTools.htm	7
V	http://www.volere.co.uk/tools.htm	71
W	http://www.atwebo.com/case.htm#Requirements%20Capture	41
Total sample size after discarding invalid tools (repeated, not in force, abandoned or renamed)		100

itative because it usually requires a subjective assessment of the relative importance of different features and how well a feature is implemented in the corresponding tool. Bearing in mind that a set of desirable features or capabilities in the tools has been identified and selected, it seems that a qualitative approach might be more suitable than a quantitative one if the study is to be accomplished. Following the DESMET terminology, a qualitative survey is a feature-based evaluation carried out by people who have experience in using or have studied the tools of interest, and it is similar to the formal experiment approach because it solicits an assessment from a number of different people, but differs in that they are not the potential users. The most qualified expert in any tool should, in theory, be its own vendor and they have therefore been found to be the most suitable respondents of the questionnaire.

A Web-based survey was created and implemented by using LimeSurvey,¹ an open source survey application. The goal of this was to make it easier for vendors to fill out the questionnaire. With regard to its design, 126 enquiries constitute the technical part of the survey, since the number of features in the classification framework (see Table 1) directly corresponds to the number of effective technical questions in the questionnaire (i.e. each question of technical nature in the questionnaire is exactly related to a certain feature in the classification framework, and therefore belongs to a certain category of features). Table 4 contains a sample of the technical questions that the vendors were asked. There was also an additional set of 20 questions in the questionnaire regarding the tools' basic administrative information (e.g. tool name, vendor name, current version). Hence, the total amount of enquiries is 146. Nevertheless, we would like to clarify that although the questionnaire was principally made up of closed questions, open questions were also included in each section of the questionnaire in order to provide the respondents with the opportunity to add comments (e.g. additional features that did not appear in the questionnaire). These questions allowed us to identify capabilities beyond the scope of our classification framework that existing tools support (see Section 5.5). For the sake of space, the fully detailed list of the capabilities explored is not shown in this manuscript. However, our RE tool survey is publicly available on a website,² which is currently evolving in order to offer extended and updated contents and information.

When answering the questionnaire, the tools' representatives had the option of not filling in the questions because they did not know the answer ("I don't know") or because of an undetermined reason ("No answer"). But this situation might lead to survey results being questioned when the amount of unanswered enquiries is too high: (1) if these unanswered questions are simply considered as negative answers (i.e. the feature is not present in the tool) then the tool results achieved may be worse than they actually are; (2) if these unanswered questions receive a positive value (i.e. the feature is fully or partially present in the tool) then, evidently, the tool results achieved may be better than they actually are. In order to minimize this potential problem, the following solution has been adopted: for each set of questions which represent the whole set of features corresponding to a concrete category, if the tool representative has answered at least 50% of the questions then the tool is considered as a participant; otherwise, the tool is not considered as a participant and the questions answered (if any) within the category are discarded. As a result, the number of participants in each category of features may vary accordingly. Moreover, we wish the level of accomplishment of the entire classification framework to be visualized through a global score. However, the unanswered questions affect the calculation of that mark for each tool. Thus, this calculation is only

performed for those tools which are considered as participants in all categories of features.

4.4. Experimental procedure

The questionnaire was prepared by the authors from September–November 2010. The six authors worked on the questionnaire design and carried out up to five revision cycles. During this time, the formulation criteria for the enquiries was carefully discussed and agreed, in an attempt to obtain questions that would be as simple and clear as possible. There was also one person in charge of technical-related tasks: survey system installation and configuration in Web server, survey creation and insertion of questions, survey conduction and system administration.

Contact with the tool representatives was always done by email, using the contact information previously collected. On December 20th, 2010 each tool representative was invited to participate in the survey by means of a request submission. This represented the first occasion on which we communicated with them and they were asked to fill out the questionnaire until January 15th, 2011. On January 8th, 2011 a reminder email was sent to each tool representative who had still not completed the questionnaire. On January 15th, 2011 the amount of answers received was 31, at the end of the predetermined time slot. A follow-up email was then sent to each tool representative who had not completed the questionnaire, asking why they had not replied to it and extending the participation deadline until February 7th, 2011 for those tool representatives who still remained interested. An additional set of three answers was then received. Finally, the authors talked directly to several vendors who had not answered the questionnaire, in order to encourage them to participate. This allowed four additional answers to be collected. The final amount of answers eventually received was 38.

After receiving the follow-up email, four tool representatives informed us why they had decided not to participate, assuming the following possibilities: (1) "Limited time to fill out the survey", (2) "Tool development abandoned", (3) "Not an RE tool as such" and (4) "Other reasons". One tool representative answered the second option, two tool representatives argued the third reason and one tool representative answered the fourth option—explaining that the information requested was confidential—. Moreover, we could not find any way in which to contact two tool representatives because their email addresses were unavailable. In conclusion, the original amount of 100 tool candidates for participation had to be reduced to 94.

4.5. Metrics

As shown previously, the features are grouped in categories of features. Since the hypotheses are linked to the categories of features, the answers to the corresponding questions obtained from the RE tools' vendors are used as a metric with which to answer each hypothesis. The metrics defined to measure the hypotheses through questionnaire scores are depicted in detail as follows. For each category of features c , the participation of the tool t in the category c is determined as follows:

$$participant(t, c) = \begin{cases} \text{true}, & NA(t, c) \geq NQ(c)/2; \\ \text{false}, & \text{otherwise.} \end{cases}$$

$NA(t, c)$ is the number of answers for the tool t in the category c , and $NQ(c)$ is the number of questions in the category c . If $participant(t, c) = \text{true}$, then the score for the tool t in the category c is calculated using the formula:

$$score(t, c) = \frac{\sum_{q=1}^{NQ(c)} score(t, q)}{NQ(c)}$$

¹ <http://www.limesurvey.org/>.

² <http://www.um.es/giisw/EN/re-tools-survey/>.

Table 4

Categories of features, their subdivisions and examples of questions. The responses permitted were “Yes”, “No”, “I don’t know” and “No answer”. EL: elicitation, AN: analysis, SP: specification, MO: modeling, VV: verification and validation, MA: management, TR: traceability, OT: other capabilities.

Category	Group	Example of question
EL	Requirements capture	The RE tool supports requirements capture by allowing the user to: store and manage the documentation for interviews, workshops, and observation
EL	Elicitation templates and checklists	The RE tool supports elicitation templates and checklists as follows: by storing and managing templates for elicitation (e.g. Quality Function Deployment (QFD) or Goal Question Metric (GQM))
EL	Importing and exporting to and from other sources	The RE tool supports importing and exporting to and from other sources as follows: by providing an OMG Requirements Interchange Format (ReqIF/RIF) compatibility for exchanging requirements information (ReqIF, formerly RIF, defines an open, non-proprietary exchange format)
EL	Elicitation documentation	The RE tool supports elicitation documentation as follows: by storing and managing textual requirements statements using basic text processor, spell and grammar checker (e.g. word spelling check, passive vs. active voice)
AN	Quality requirements analysis	The RE tool supports quality requirements analysis as follows: by storing and managing quality requirements in quality attributes, policies, or constraints
AN	Feasibility analysis	The RE tool supports feasibility analysis as follows: by storing and generating checklists or templates for various analyses (e.g. technical, economical, and operational analyses)
AN	Attribute analysis	The RE tool supports the tracking of any attribute(s) user defined or tool provided as follows: by detecting and flagging missing attributes
AN	Risk analysis and management	The RE tool supports risk analysis and management as follows: by maintaining relationship of requirements to risks raised and risks mitigated
SP	Requirements specification documentation	The RE tool supports requirements specification documentation as follows: by synchronizing changes between the RE tool and formatted document (complete loop between RE tool and formatted document)
MO	Modeling analysis	The RE tool supports modeling analysis as follows: by evaluating requirements based on business goals
MO	Modeling and specification languages	The RE tool supports the storage and display of: natural language statements
VV	Verification and validation	The RE tool supports V&V as follows: by generating exception reports on requirements that do not have verification plan cases and verification plan cases that are not linked to requirements
MA	Baseline of the requirements	The RE tool supports baseline of the requirements as follows: by generating reports that compare and contrast baselines versions
MA	Requirements change management	The RE tool supports requirements change management as follows: by generating requirements modification history reports that should include, change author, change id, date of change, change status such as accepted, rejected or pending
MA	Project management	The RE tool supports project management as follows: by recording, tracking, and reporting the status of the overall requirements management process
MA	Open or closed data model	The RE tool supports an open data model as follows: by providing a standard format for important application data structures, which is completely accessible to all external programs while the application is running
TR	Traceability	The RE tool supports traceability as follows: by automatically maintaining the traces
TR	Flexible tracing	The RE tool supports flexible tracing as follows: forward and backward tracing
TR	Bi-directional tracing	The RE tool supports bi-directional tracing of: requirements to child requirements or design elements
TR	Traceability analysis	The RE tool supports traceability analysis as follows: by generating traceability reports to identify exceptions in user requirements in the analysis phases to functional requirements
OT	RE tool administrative information	The RE tool supports this information as follows: by providing project information (e.g. project size, concurrent users, and number of analyst)
OT	Graphical user interface	The RE tool supports graphical user interface as follows: by enabling Web browser interface
OT	Data integration	The RE tool supports data integration by using: database federation to provide users with a virtual data warehouse which removes the need to move any of the data, provides access to “live” data and functions, and efficiently combines data from multiple sources of different types by using a single arbitrarily complex query

$score(t,q) \in \{0,1\}$ is the score for the tool t in the question q and $score(t,c) = s \in [0,1]$ is then discretised on a 5-interval scale using a global unsupervised discretisation method [32], a variation of the equal width interval binning in which the lower and upper bins are shorter than the others with the intention of discriminating between extreme scores:

$$discretise(s) = \begin{cases} \text{Very high,} & s \in (0.875, 1]; \\ \text{High,} & s \in (0.625, 0.875]; \\ \text{Medium,} & s \in (0.375, 0.625]; \\ \text{Low,} & s \in (0.125, 0.375]; \\ \text{Very low,} & s \in [0, 0.125]. \end{cases}$$

The score for the tools in the category c is determined as follows:

$$score(c) = \frac{\sum_{t=1}^{NP(c)} score(t,c)}{NP(c)}$$

$NP(c)$ is the number of participants in the category c . In order to determine the truth or falsity of the hypotheses posed in Section 4.2, we shall consider that each hypothesis **H1.1** to **H1.9** is true if

the score of the tools in the corresponding category of features is greater than or equal to 0.70, i.e. if $score(c) \geq 0.700$. Otherwise we consider it false. We have chosen such a high threshold for accepting or rejecting the hypotheses with the intention of ensuring that the tools provide the features analysed with proper support.

The calculation of the values corresponding to the cost per individual license and the number of licenses in use is slightly different from the method explained previously owing to their different nature. Furthermore, the following formulae are applied in order to obtain quantitative values from the ranged values which were gathered using the questionnaire (i.e. by passing from price p or number of licenses l intervals into scores $s' \in \{0.875, 0.625, 0.375, 0.125, \text{No answer}\}$):

$$quantify(p) = \begin{cases} 0.875, & p = \text{More than 1000;} \\ 0.625, & p = 501 \text{ to } 1000; \\ 0.375, & p = 100 \text{ to } 500; \\ 0.125, & p = \text{Less than } 100; \\ \text{No answer,} & p = \text{No answer.} \end{cases}$$

Table 5
Participants.

No.	Tool	Vendor	Web
1	Acclaro DFSS	Axiomatic Design Solutions, Inc.	http://www.dfss-software.com/
2	Aligned Elements	Aligned AG	http://www.aligned.ch/Product/Product.html
3	Avenqo PEP	Avenqo, Germany	http://www.avenqo.com/index.php/en/avenqo-pep
4	Blueprint	Blueprint Software Systems Inc.	http://www.blueprintsys.com/product_rc.php
5	Bright Green Projects	Bright Green	http://www.brightgreenprojects.com
6	Caliber RM	Micro Focus	http://www.borland.com/us/products/caliber/index.html
7	Cameo Requirements+	No Magic, Inc.	http://www.magicdraw.com/cameoreq
8	CASE Spec	Goda Software	http://www.casespec.net/requirementsmanagement.htm
9	Cognition Cockpit	Cognition Corporation	http://www.cognition.us/cockpit_overview.html
10	Cradle	3SL	http://www.threesl.com/pages/products/index.php
11	G-MARC	Computer System Architects Ltd.	http://www.informeng.com/
12	inteGREAT	eDev technologies	http://www.edevtech.com
13	IRqA	Visure Solutions	http://www.visuresolutions.com/web/visure/requirementsdefinitionmanagement
14	jUCMNav	jUCMNav	http://jucmnav.softwareengineering.ca/ucm/bin/view/ProjetSEG/WebHome
15	Leap SE	Leap Systems	http://www.leapse.com
16	MacA&D/WinA&D	Excel Software	http://www.excelsoftware.com/matrix.html
17	MKS Integrity	MKS Inc.	http://www.mks.com/solutions/discipline/rm
18	PACE	ViewSet Corporation	http://www.viewset.com/index.php/products-pace-overview
19	Polarion Requirements	Polarion Software	http://www.polarion.com/products/requirements/index.php
20	Psoda	Psoda	http://www.psoda.com/cms.php/what-is-psoda/requirements-management
21	QFDcapture	International TechneGroup Incorporated	http://www.qfdcapture.com/
22	QPack	Orcanos	http://www.orcanos.com/Requirements_management.htm
23	RaQuest	SparxSystems Japan	http://www.raquest.com/
24	Rational DOORS	IBM Rational	http://www-01.ibm.com/software/awdtools/doors/
25	ReqMan	RequirementOne Inc.	http://www.requirementone.com
26	Reqtify & Requirement Central	Dassault Systemes	http://www.reqtify.com
27	Requirements Composer	IBM Rational	http://www-01.ibm.com/software/awdtools/rrc/
28	RTIME	QAvantage	http://qavantage.toolsforproductmanagement.com/
29	RequisitePro	IBM Rational	http://www-01.ibm.com/software/awdtools/reqpro/
30	RMTrak	Prometeo Technologies	https://www.rmtrak.com/req_management.aspx
31	Rommana	Rommana Software	http://www.rommanasoftware.com/
32	Scenario Plus	Scenario Plus	http://www.scenarioplus.org.uk/
33	SpiraTeam	Inflectra Corporation	http://www.inflectra.com/SpiraTest/Default.aspx
34	TestTrack RM	Seapine Software	http://www.seapine.com/ttrm/requirements-management-software
35	TopTeam Analyst	TechnoSolutions Corp	http://www.technosolutions.com/topteam_requirements_management.html
36	TraceCloud	TraceCloud	http://www.tracecloud.com
37	TrackStudio	TrackStudio	http://www.trackstudio.com/
38	VisibleThread On-premise/On-demand	VisibleThread	http://www.visiblethread.com

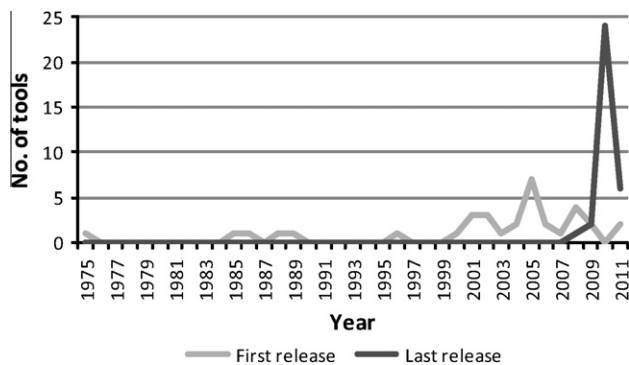


Fig. 1. Year of first and last release.

$$\text{quantify}(l) = \begin{cases} 0.875, & l = \text{More than } 10,000; \\ 0.625, & l = 1001 \text{ to } 10,000; \\ 0.375, & l = 101 \text{ to } 1000; \\ 0.125, & l = 0 \text{ to } 100; \\ \text{No answer}, & l = \text{No answer}. \end{cases}$$

4.6. Analysis procedure

The Pearson’s Correlation Coefficient was used to discover relevant associations between variables, since this measure determines

the strength of the linear relationship between two quantitative variables [33]. It is easy to work out and to interpret, and is therefore widely used. The data collected were analysed and figures were generated by employing the SPSS 19.0 statistical software package and Microsoft Office Excel 2007. The interrater reliability analysis was performed by using Analyse-it for Microsoft Excel (version 2.20).³

5. Results

In this section, the participants of the study are depicted from the administrative information collected in the survey, the correlation tests between the variables defined in this study are provided, the hypotheses of the study are evaluated, the procedure to validate the data is presented, and finally, the discussion of the results is supplied.

5.1. Participants

There were 38 participants (see Table 5) out of 94 candidates invited, which signifies that a 40.4% participation rate was achieved. With regard to the administrative questions in the questionnaire, and before studying the hypotheses posed in Section 4.2, a general data analysis was carried out including the tool release

³ www.analyse-it.com/.

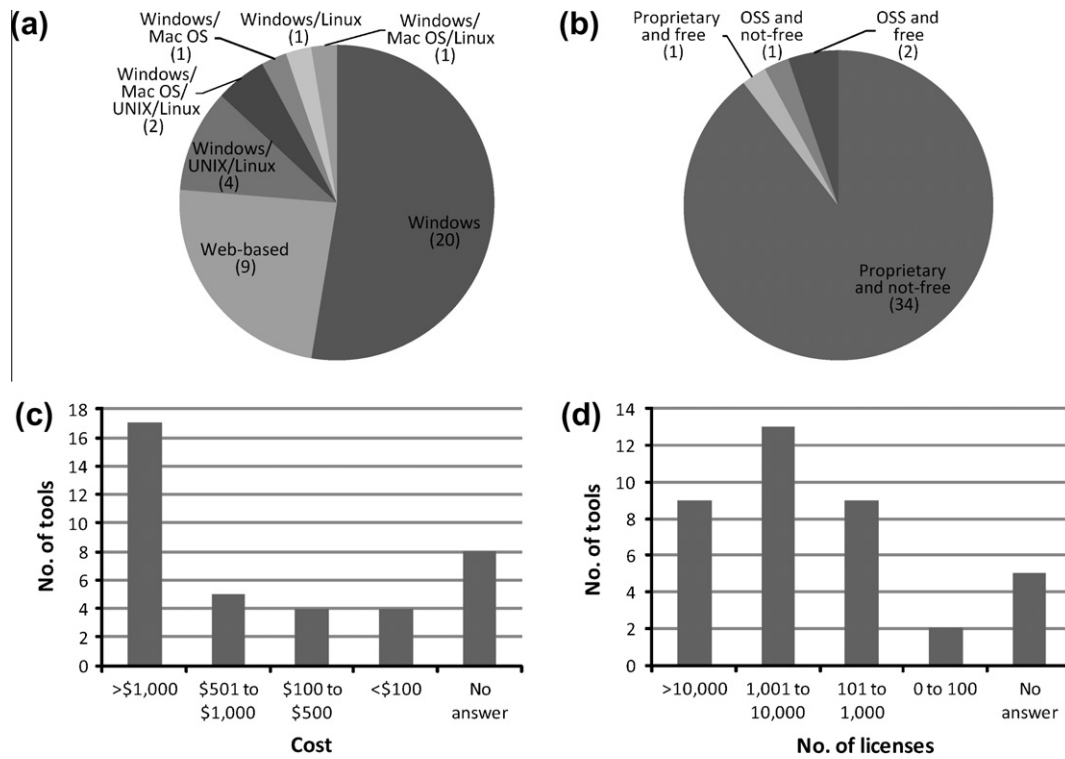


Fig. 2. (a) Platform required, (b) type of license, (c) cost per individual license and (d) number of licenses in use.

date, the operating system required, the license type, the cost per individual license and the number of licenses in use (see Fig. 1 and Fig. 2). A 34 out of the 38 tool representatives who took part in the survey answered the question about the year of the first release of the tool and 33 answered the question about the year of the last release. Most participants reported that their tools were re-

leased for the first time in the last decade and these tools seem to be very updated (Fig. 1). With regard to the software platform required, there is a great predominance of Windows systems, as shown in Fig. 2a. Web-based clients are also common, on the road to facilitating collaborative access to resources. Other operating systems such as UNIX or Linux, and particularly Mac OS, have a

Table 6

Correlation matrix. EL: elicitation, AN: analysis, SP: specification, MO: modeling, VV: verification and validation, MA: management, TR: traceability, OT: other capabilities, GL: global, CO: cost per license, Li: licenses in use. N: sample size.

	EL	AN	SP	MO	VV	MA	TR	OT	GL	CO	Li
EL	1										
N	35										
AN	0.763**	1									
N	34	35									
SP	0.800**	0.653**	1								
N	35	35	36								
MO	0.470**	0.499**	0.629**	1							
N	33	33	34	34							
VV	0.792**	0.740**	0.641**	0.521**	1						
N	27	27	27	25	29						
MA	0.729**	0.725**	0.716**	0.592**	0.694**	1					
N	31	31	32	31	28	34					
TR	0.776**	0.618**	0.781**	0.656**	0.791**	0.737**	1				
N	31	31	32	30	29	33	34				
OT	0.777**	0.610**	0.718**	0.628**	0.678**	0.801**	0.732**	1			
N	29	29	29	28	28	31	31	31			
GL	0.903**	0.799**	0.869**	0.799**	0.830**	0.934**	0.913**	0.872**	1		
N	25	25	25	25	25	25	25	25	25		
CO	0.269	0.336*	0.545**	0.404*	0.169	0.094	0.329*	0.147	0.358	1	
N	29	29	30	28	23	27	27	24	21	30	
Li	0.185	0.068	0.285	-0.006	-0.054	0.245	0.012	0.513**	0.183	0.243	1
N	31	31	32	30	25	30	30	27	22	29	33

* Correlation is significant at the 0.05 level (1-tailed).

** Correlation is significant at the 0.01 level (1-tailed).

more limited presence. Moreover, the tools' licenses are mainly proprietary and not-free (see Fig. 2b), with a low influence of open-source software (OSS). Regarding the average cost per individual license (in US dollars), most tools cost \$1000 or more as shown in Fig. 2c, and up to eight tool representatives did not answer the question. Finally, the number of licenses in use is commonly between 1001 and 10,000 (Fig. 2d), but there is an amount of more extended tools (more than 10,000 licenses) and another important group of tools with less representation in the market (between 101 and 1000 licenses).

5.2. Correlation between variables

Bivariate correlation tests have been applied to measure the strength of linear dependence between the aforementioned variables. Table 6 shows the results attained. To clarify its meaning, each cell contains two numbers: the first is the *r-value*, which indicates the strength and direction (\pm) of the correlation (bigger is better) and the “*” or “***” signifies that the null hypothesis H_0 can be rejected (variables really correlate between them). The second is the number of pairs in the sample.

High positive values representing strong positive correlation between variables were obtained when considering the distinct categories of features (e.g. elicitation with analysis, 0.763; elicitation with specification, 0.800) thus scores achieved in each distinct category of features are connected with all the other categories of features.

Moreover, positive correlation has been uncovered between the average cost per individual license and (1) the analysis features (0.336), (2) the modeling features (0.404), and (3) the traceability features (0.329) offered by the RE tools addressed in this study. In addition, there is an even stronger positive correlation between the average cost per individual license and the tool support to the requirements specification features (0.545). Nevertheless, when looking for an association between the average cost per individual license and the global score, the correlation result is not significant although it is positive (0.358), so we cannot conclude that such a connection exists.

Finally, the correlation between the number of licenses in use and the remaining variables was also calculated. The results reveal a strong positive correlation between the number of licenses in use and the other capabilities category (0.513). However, there is no significant correlation between the number of licenses in use and the average cost per individual license, even though the value is positive (0.243). Furthermore, the correlation between the number of licenses in use and the global score has a positive value (0.183) but this value is not sufficiently high to be significant.

5.3. Evaluation of hypotheses

A descriptive statistical approach was used to test the hypotheses of this study. As is shown in Fig. 3, the vendors' answers to all questions, grouped by category of features, have been incorporated into the same graph to be compared with each other.

Quite a good level of effectiveness can be observed when considering the analysed features all together. However, there was a significant difference among tools' effectiveness by category:

- A total of 30 tools obtained *High* or *Very high* scores in the requirements elicitation category (85.7% of the participants in this category of features), which confirms that this aspect is widely supported by current tools. **H1.1** is accepted since the score achieved by the tools is 0.80, standard deviation (SD) = 0.23 (see Section 4.5).
- In total, 26 tools achieved *High* or *Very high* scores in the requirements analysis category (74.3%). It is possible to affirm that most of the RE tools under study adequately support these features (0.76, SD = 0.23) and **H1.2** can be accepted.
- 27 tools rated *High* or *Very high* in requirements specification (75.0% of the participants). A score of 0.73, SD = 0.27 was achieved by the tools, and **H1.3** is thus accepted.
- A total of 15 tools obtained *Medium*, *Low* or *Very low* scores in the requirements modeling category of features (44.1%). This result does not support **H1.4**, and it can therefore be rejected (0.59, SD = 0.32).
- 23 out of 29 tools (79.3%) satisfactorily supported the requirements V&V features with *High* or *Very high* scores in this category. Moreover, the score of the tools (0.79, SD = 0.28) confirms **H1.5**.
- 14 tools (41.2%) achieved *Medium*, *Low* or *Very low* scores in the requirements management category. In addition, the score attained by the tools in this case is 0.67, SD = 0.28, signifying that **H1.6** can be rejected.
- 26 tools rated *High* or *Very high* in requirements traceability (76.5% of the participants). Furthermore, the score of the tools is 0.78, SD = 0.23. Hence, **H1.7** is accepted.
- It is also noticeable that the participants adequately support the other capabilities category, with 22 out of 31 tools attaining *High* or *Very high* scores (71.0%). Moreover, **H1.8** can be accepted given that the score achieved by the tools is 0.71, SD = 0.26.
- In general, quite a good level of effectiveness was observed in the tools analysed (see Fig. 4). 72.0% of the tools rated *High* or *Very high* and their score in all the categories of features as a whole was 0.72, SD = 0.24. **H1.9** is therefore accepted.

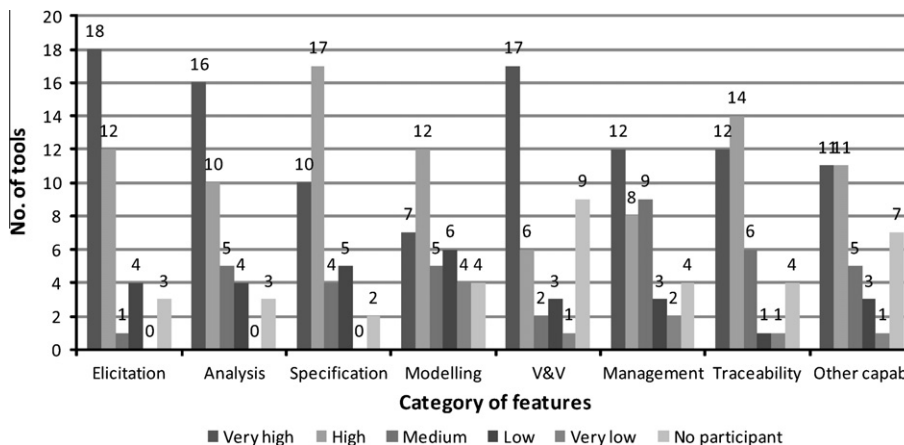


Fig. 3. Tools' score.

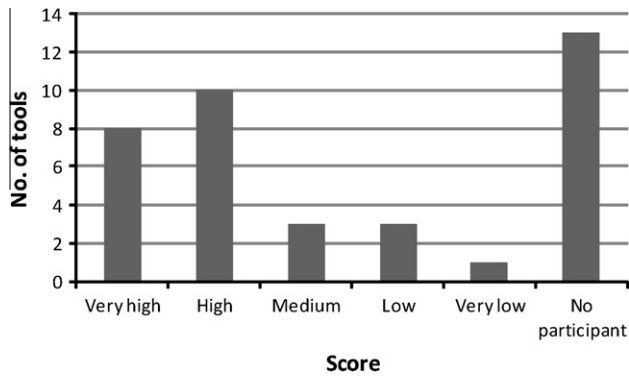


Fig. 4. Tools' global score.

5.4. Validation of the data

A sample of the tools included in the survey was randomly selected and subjected to neutral assessment, since an evaluation made exclusively by the vendors might bias the results owing to a lack of objectivity. An interrater reliability analysis was then performed with the purpose of determining the consistency among raters and measuring the reliability of the data gathered from the vendors' answers. More specifically, we meticulously assessed the capabilities of 5 tools out of 38 participants (13.2%) and cross-checked them against the vendors' evaluation of their tools.

Interrater agreement is commonly used to evaluate the agreement between different classifications, which might include nominal or ordinal scales. In particular, the Cohen's Kappa coefficient [34] makes it possible to calculate this agreement between two observers on the assignment of classes of a categorical variable, although it does not take into account the degree of disagreement between observations. Since we start from ordered codes (*Very high*, *High*, *Medium*, *Low* and *Very low*) and Weighted Kappa [35] allows different levels of agreement to contribute to the value of Kappa when the categories are ordered (i.e. when the variable is ordinal), this method was eventually applied. Quadratic weights were also selected rather than linear weights, as the difference between the first and second category is less important than the difference between the second and third category, etc. The interrater reliability for the observers was found to be $Kappa = 0.63$ ($p < 0.0001$), 95% CI (0.4, 0.85). The strength of agreement is therefore *Substantial* [36] or *Good* [37], which indicates that the data obtained seem to be trustworthy. The p-value and confidence interval were calculated using the Fleiss method [38].

We have raised the level of confidence achieved by using the triangulation technique [39] in the cross-check, in order to ensure that the data collected enables the researchers to draw valid conclusions. Three external assessors tested the tools (one assessor was assigned to each tool), three researchers supervised the evaluation work, and another three researchers reviewed their findings. A comparison was eventually made to ensure similar conclusions.

5.5. Discussion

In the following paragraphs, we discuss the research contributions of this study, the practical implications of our findings and the threats to the validity of the study.

5.5.1. Research contributions

We have searched scientific literature, in order to discover other pieces of research that have analysed important features that should be supported by RE tools, with the purpose of addressing whether or not our results are aligned with previous researchers' findings and complemented by previous studies:

- There is a lack of updated studies on RE tool's capabilities and comparative frameworks. In addition, existing studies are focused on specific RE activities (e.g. elicitation, V&V) and analyse a reduced amount of RE tools (at the very most 10), whereas this manuscript is focused on the entire RE process and we have used a survey approach to extend the scope of participants.
- Other studies agree with ours in that there is a lack of information about RE tools and information gathering requires great effort [22], along with the fact that a certain degree of subjective interpretation that is difficult to mitigate is still possible even though in principle the evaluation criteria for assessing RE tools may seem to be well defined [24]. Like us, Schmid et al. [22] also experienced a high diversity of responsiveness when contacting the tool vendors.
- Schmid et al. [22] observed striking differences among RE tools in the area of supporting V&V. We have reported a mainly good level of support for these features, although this category of features suffers the highest rate of non-participation. Zowghi and Coulin [23] claim that a wide variety of tools have been developed and used to support elicitation, regardless of whether or not they have been specifically designed for it, which may be related to our findings on the relative maturity of the RE tools as regards elicitation features. Beuche et al. [24] took the requirements catalogue for single product developments written by Homann et al. [2] as a basis, and adapted it in order to make it suitable for product line development, while Toval et al. [15] depicted the features needed for their reuse-centered RE tool. In both cases, the results of their evaluation are highly biased towards product lines features and requirements reuse features respectively, and it is not therefore possible to compare these results with those achieved in our study.

With regard to the RE tools' evaluation environment, it is important to note that:

- This study attempted to characterize the relevant RE tools' features by means of a specific classification framework, which is mainly based on the ISO/IEC TR 24766 guideline in order to provide formal backing, but also contains additional capabilities.
- One important concern for us was the completeness of the ISO/IEC TR 24766 framework, since important tool features might be overlooked despite our efforts in the design of the study. As stated previously, the questionnaire was based on the ISO/IEC TR 24766 guideline but open questions were included in order to discover additional tool capabilities beyond its scope. Some examples of these new features are outlined briefly as follows:
 - Analysis of natural language semantics in addition to grammatical and morphological analysis to ensure objectivity in requirements expression.
 - Storage and display of embedded (rather than simply attached) diagrams and graphics to grant flexibility in being able to load, manage and display virtually any model and language.
 - Provision of an application programming interface (API) to facilitate advanced integration capabilities.

Based on the results of the correlation tests, we have demonstrated that:

- The strong positive correlation uncovered between the distinct categories of features suggests that when an RE tool is good in an RE activity (e.g. elicitation), then this tool is usually good in the other RE activities. Moreover, when an RE tool is weak for an RE activity, then it is usually weak for the others as well.

- The positive correlation found between the average cost per individual license and the analysis, modeling, traceability and specification features indicates that the more expensive a tool is, the better requirements analysis, modeling, traceability and specification support it provides. Thus, strong investment in RE tools is encouraged when those categories of features are required to be present. Since correlation between the average cost per individual license and the global score achieved by the tools has not been found, RE tools' customers should undertake an in-depth research into those applications that are not so expensive, since they may find a satisfying solution in light of the results.
- The statistical significance of the correlation between the number of licenses in use and the other capabilities suggest that users have an interest in capabilities that do not necessarily fall into any particular process identified in the ISO/IEC TR 24766, but complement the traditional RE activities. Particularly, the other capabilities includes features regarding RE tool administrative information, graphical user interface and data integration. In addition, a connection between the number of licenses in use and the average cost per individual license did not arise, which indicates that more expensive tools are not better sold than cheaper ones in the RE tools' marketplace. The absence of a significant correlation between the number of licenses in use and the global score similarly highlights that the most extended RE tools are not always the best solutions.

With regard to the hypotheses evaluation, the descriptive statistical analysis has shown that:

- Requirements elicitation exemplifies the best supported category of features in this study (**H1.1**).
- Requirements analysis, specification and traceability represent the second group of best supported categories of features (**H1.2**, **H1.3** and **H1.7**).
- Requirements V&V and other capabilities are the following categories of features with regard to their level of support (**H1.5** and **H1.8**).
- Requirements modeling and management are the most badly supported categories of features in this study (**H1.4** and **H1.6**).
- The global score is sufficiently high to affirm that the RE tools perform well in the whole RE process (**H1.9**). Nevertheless, an important amount of tools have been considered as non-participants in this scale owing to the calculation method selected, as shown in Fig. 4. In fact, only participants in all categories of features are suitable to be included in this measure.

5.5.2. Practical implications

Our research directs readers to understand and measure different dimensions of the RE tools under study. The level of accomplishment of the different groups of capabilities that belong to the eight categories of features and the number of features in each one, might particularly help both researchers and practitioners to reflect on which features are better supported and which others are not so well covered by current RE tools. This information is shown in Table 7.

Requirements elicitation is the best supported category of features in this study. Moreover, requirements capture is one of the

Table 7
Categories of features, their subdivisions, number of features in these groups, percentage of tools which support these features and examples of features. EL: elicitation, AN: analysis, SP: specification, MO: modeling, VV: verification and validation, MA: management, TR: traceability, OT: other capabilities.

Category	Group	No.	Pct.	Example of feature
EL	Requirements capture	12	80.7	Store and manage the documentation for interviews, workshops, and observation
EL	Elicitation templates and checklists	3	58.8	Store and manage templates for elicitation (e.g. Quality Function Deployment (QFD) or Goal Question Metric (GQM))
EL	Importing and exporting to and from other sources	3	63.2	Provide an OMG Requirements Interchange Format (ReqIF/RIF) compatibility for exchanging requirements information
EL	Elicitation documentation	2	73.7	Store and manage textual requirements statements using basic text processor, spell and grammar checker (e.g. word spelling check, passive vs. active voice)
AN	Quality requirements analysis	2	67.1	Store and manage quality requirements in quality attributes, policies, or constraints
AN	Feasibility analysis	2	69.7	Store and generate checklists or templates for various analyses (e.g. technical, economical, and operational analyses)
AN	Attribute analysis	5	76.8	Detect and flag missing attributes
AN	Risk analysis and management	2	60.5	Maintain relationship of requirements to risks raised and risks mitigated
SP	Requirements specification documentation	7	69.2	Synchronize changes between the RE tool and the formatted document (complete loop between RE tool and formatted document)
MO	Modeling analysis	4	59.9	Evaluate requirements based on business goals
MO	Modeling and specification languages	9	50.3	Store and display natural language statements
VV	Verification and validation	11	64.6	Generate exception reports on requirements that do not have verification plan cases and verification plan cases that are not linked to requirements
MA	Baseline of the requirements	4	67.1	Generate reports that compare and contrast baselines versions
MA	Requirements change management	7	64.7	Generate requirements modification history reports that should include, change author, change id, date of change, change status such as accepted, rejected or pending
MA	Project management	2	71.1	Record, track, and report the status of the overall requirements management process
MA	Open or closed data model	5	49.5	Provide a standard format for important application data structures, which is completely accessible to all external programs while the application is running
TR	Traceability	10	72.1	Automatically maintain the traces
TR	Flexible tracing	7	57.9	Provide forward and backward tracing
TR	Bi-directional tracing	6	81.1	Trace of requirements to child requirements or design elements
TR	Traceability analysis	6	73.7	Generate traceability reports to identify exceptions in user requirements in the analysis phases to functional requirements
OT	RE tool administrative information	5	62.1	Provide project information (e.g. project size, concurrent users, and number of analyst)
OT	Graphical user interface	6	77.6	Enable Web browser interface
OT	Data integration	6	41.2	Use of database federation to provide users with a virtual data warehouse which removes the need to move any of the data, provides access to "live" data and functions, and efficiently combines data from multiple sources of different types by using a single arbitrarily complex query

best supported groups of features. Despite this, there is a lack of support for elicitation templates and checklists. Hence, although the requirements capture is well covered, it is important to additionally provide a consistent structure with which to transcript the requirements and other related information.

Requirements analysis, specification and V&V are also satisfactorily supported categories of features. The strong support for the attribute analysis group of features, which allows metrics associated with requirements to be defined and kept track of is notable. On the other hand, better support for risk analysis and management features is missing. However, this situation should not be cause for concern, since such features are perhaps more related to project management than to requirements analysis.

Requirements modeling is one of the categories of features most poorly supported by the RE tools. Particularly, the support given to the different modeling and specification languages has been revealed to be one of the most badly supported groups of features in this study. Table 8 shows that SysML artifacts, goal models, BPMN and DFDs are, in this order, the key features that stand out as the most suitable for improvement, whereas natural language statements, and to a lesser degree, context diagrams, are usually well covered by the RE tools.

The different groups of features within the requirements management category of features are in general satisfactorily supported. However, the open data model is the most badly supported group of features in the study, and some efforts might therefore be made to provide such capabilities (see Table 9). Among other mechanisms, the RE tools might improve their support for “scriptability”, and enable external applications to monitor and execute operations provided to operate on the data, since they are the most poorly supported features in this group. Likewise, a standard format for important application data structures, which is completely accessible to all external programs while the application is running, is often provided by the RE tools.

Overall, requirements traceability has proved to be very well supported by the RE tools. Bi-directional tracing stands out as the best supported group of features in this study. Only the group of features relative to flexible tracing is clearly below the other groups, thus inviting tool improvement in order to allow tracing between different elements (e.g. text, graphics, tables).

The other capabilities, such as the requirements management category of features, has a quite good level of accomplishment but fails in one of its groups of features: data integration. Moreover, data integration is the most badly supported group of features in this study. Furthermore, Table 10 shows an almost decreasing trend with regard to the degree of support that the RE tools give to different data integration mechanisms, which is coherent with the complexity of the means of data integration stated.

Finally, as mentioned in Section 5.3, there is a significant imbalance among RE tools' scores by category of features. Nevertheless, Table 7 has shown that such an imbalance is occasionally even greater within the categories themselves, i.e. between the different

Table 8
Support modeling and specification languages.

Name	Pct.
Natural language statements	71.1
Context diagrams	60.5
Conceptual domain models	50.0
Business Process Modeling Notation (BPMN)	44.7
Goal models	39.5
Systems Modeling Language (SysML) artifacts	31.6
Unified Modeling Language (UML) artifacts	55.3
Data Flow Diagrams (DFDs)	44.7
Entity/Relationship (E/R) diagrams	55.3

Table 9
Support an open data model.

Name	Pct.
External applications can record the user's actions and execute any recorded command (“scriptability”)	36.8
External programs can examine and modify the data structures of the application (“examinability”)	52.6
Standard format for important application data structures is provided, which is completely accessible to all external programs while the application is running	60.5
Standards are provided for how operations and their parameters and results are represented	55.3
External applications are enabled to monitor and execute operations provided to operate on the data	42.1

Table 10
Support to data integration.

Name	Pct.
Application-specific solutions	57.9
Application-integration frameworks	55.3
Workflow (or business process integration) frameworks	39.5
Digital libraries with portal-style/meta-search engine integration	29.0
Data warehousing	36.8
Database federation	29.0

groups of features. This is owing to the considerable organizational autonomy inherent to the different groups of features, even though they are classified into categories of related features.

5.5.3. Research limitations

The threats to the validity [29] of this research are discussed below.

Internal validity. This is concerned with the reliability of the results. The validity of the material gathered through the questionnaire is highly dependent on the experience of the respondents. Most of the participants were senior personnel in their organizations, and therefore had several years' experience with the tool. The risk of maturation was taken into account by ensuring that the questionnaire took no more than 20 min to complete. In spite of this, there was a mortality percentage of 7.3% (3 out of 41 respondents began to complete the survey, but then abandoned it part way through). The effect of a low interest in or commitment to the questionnaire showed by some tool representatives, owing to their desire to complete the question-answering task quickly, mistrust toward interviewers, ignorance of the topics being asked about or whatever other circumstances might be a serious problem. The reliability of these incoherent or incomplete answers has been questioned and, as they produce low-quality data, they have been directly discarded to minimize their influence in the study. A further difficult issue is that of guaranteeing the truthfulness of the data, since the tools' representatives answered the questionnaire in the awareness that they were being observed and observational techniques always run the risk of changing the process simply by observing it. The Hawthorne effect [40] might therefore have led the vendors to deliberately skew their answers in a particular direction. With regard to its mitigation: (1) careful consideration of this effect is warranted in implementing the research and explaining its purpose and protocol to the research participants [41], and (2) the execution of an interrater reliability analysis showed quite a successful *Substantial* strength of agreement, and the results obtained therefore seem to be dependable (see Section 5.4).

Conclusion validity. This refers to the ability to draw correct conclusions about relationships. The sample size (38 tools) was below that needed to produce an acceptable statistical power and it is

undeniable that a certain amount of tools not considered in this research exists. Therefore, conclusions derived from the study must not be considered definitive and further prospective research with a larger sample is recommended to explore the capabilities of current RE tools at greater length. Nevertheless, the number of participants represents an important percentage of the entire RE tool vendor community, including tools from different companies located in three continents (Asia, America and Europe). In addition, the study has been validated through a systematic process and sufficient details have been included to allow the process to be reproduced. However, the number of results obtained in the searches might be different in the future.

External validity. This is concerned with the generalization of the results to industrial practice. In order to be systematic, only tools appearing in well-known RE tool lists were included in this survey, thus omitting other tools which might be RE tools but have not yet been collected in these sources. The selection of the participants was therefore adequate, as all of the well-known RE tool databases were screened (Table 3). Unfortunately, several major tools are missing in the study, even though we tried all possible means to include them. Enterprise Architect is an example of a vendor who did deliberately not participate. In addition, the capabilities under study were mainly extracted from a framework specifically designed by experts for the evaluation of RE tools, which was also refined in order to address some additional topics linked with the needs of industry (e.g. ReqIF, open data model, data integration). However, these features might not be those that users of tools find relevant. We believe that the experimental evidence can be generalized to specific RE industrial settings, and findings from our study can be used by requirements engineers, bearing in mind that the technology is evolving and new tools will be introduced to support RE characteristics, whilst existing tools will provide new abilities.

6. Conclusions and future work

We believe that this article is relevant for researchers, since they will be able to gain an insight into the state-of-the-art on RE tools, and also for practitioners, as it will help them to be aware of the features that the current RE tools often provide, their cost, the number of licenses in use, etc. This information might be of great interest when deciding which RE tool should be used in their organizations.

This survey of RE tools has shown that, in general, the participants are strong in the great majority of the RE process activities, taking as our starting point the ISO/IEC TR 24766. Nevertheless, some enhancements could be made to current RE tools in order to provide specific capabilities that are not yet sufficiently supported. Many of these tools are located in the following categories of features: (1) requirements modeling, (2) requirements management, and (3) other capabilities. These would thus appear to be suitable areas for RE tool improvement.

Not surprisingly, the RE tool scores concerning modeling are lower than those regarding specification and other categories. We believe that this may be owing to the fact that RE tools are traditionally more oriented towards textual, natural language requirements than to modeling notations such as BPMN, UML or E/R. The modeling and specification languages that are currently supported by the tools under study confirm this theory, as is shown in Table 8.

With regard to the requirements management category, a lack of open data model mechanisms has been detected, therefore reducing the scores achieved by the tools. It would appear that current RE tools do not widely support such features, as is shown in Table 9, although they provide both developers and users with

many important benefits: support for increased communication and automation, extensive end-user customization, scripting and macro capabilities, intelligent external agents and tutors, rich search and replace commands, easy supply of significantly extended spell-checkers, semantic markings, alternative interfaces without reimplementation, the ability to have plug-ins that operate in the same space, and a significantly higher reuse of common code for implementers [42]. Some participants who have mentioned being able to support the open data model have pointed out that they achieve this goal by means of a specific API.

Sometimes, particularly in large organizations, there are various data sources containing the company's critical data. Moreover, the management of such scattered data depends on different systems. This diversity of data sources is caused by many factors which are typically found in GSD projects, including lack of coordination among different parts of the organization, different rates of adopting new technology, mergers and acquisitions, and geographic distance between collaborating groups [43]. RE tools should therefore offer mechanisms with which to combine the information from these various systems (see Table 10), above all when the work environment is distributed.

In future work we will extend the scope of the study to cover the support that current RE tools provide in GSD environments, guided by the ISO/IEC TR 24766 and scientific literature. Further and more in-depth research into concrete RE capabilities will also be carried out.

Acknowledgments

Grateful thanks are owed to the 38 participants of the survey for their invaluable help in this study. This work has been funded by the PEGASO/PANGAEA Project (TIN2009-13718-C02-02), the ORIGIN Integrated Project (IDI-2010043 (1-5)) and the ENLOBAS Project (PII2109-0147-8235).

Appendix A. Classification framework details

Concrete examples of capabilities extracted from the eight categories of features of the survey are shown as follows, in addition to the subdivisions and meaning of the categories themselves, with the intention of shedding light on the kind of capabilities that are included in each one.

- **Elicitation.** This category includes features focused on the ability of the tools to support the identification of stakeholders, the capturing and tracing of the business/user requirements, functional requirements, and the quality (non-functional) requirements during elicitation work.
 - **Requirements capture.** This group of capabilities aims to depict the support provided by the tools during the process of identifying the customer needs for the proposed system. Some of the subjects tackled are the management of stakeholders' information, the creation of hierarchical relationships between requirements, and the use of specific reporting facilities.
 - **Elicitation templates and checklists.** Both templates and checklists provide a consistent structure with which to record the requirements descriptions and other requirements related information. The management of prioritization forms is another topic included in this group of features.
 - **Importing and exporting to and from other sources.** The elicited requirements should be imported from, or interfaced to users, hardware, and other software systems. The import and export to and from other tools and standard file formats are interesting properties which might be offered by the

tools regarding this subject. In particular, since companies rarely work on the same requirements repository and do not usually work with the same RE tools [44], a standard format for requirements information is needed. ReqIF [44] is an emerging OMG exchange format that is intended to be generic, open and non-proprietary, which represents a successful step towards bridging the gap. It allows different companies to exchange requirements information across organizational boundaries without losing the advantages of managing requirements internally, thus ensuring consistency, reducing defects, speeding up information exchange, enabling collaboration and lowering costs [45].

- **Elicitation documentation.** The output from the entire requirements elicitation tasks should be stored, retrieved, and edited in various formats. This includes textual requirements statements and non-textual requirements such as graphics, tables or equations.
- **Analysis.** This category includes capabilities aimed at decomposing high-level requirements into details, evaluating feasibility, negotiating priorities, identifying conflicts, determining unclear, incomplete, ambiguous or contradictory requirements and resolving all these issues.
 - **Quality requirements analysis.** This group encompasses support features for significant architectural and design decisions. The topics addressed are storing and managing quality requirements in different manners (e.g. attributes, policies or constraints) and the rationale of quality attributes trade-off.
 - **Feasibility analysis.** This tackles features with which to evaluate the possibility of implementing each requirement at an acceptable cost and performance, and identify technical obstacles. The tools' capabilities aimed at storing and generating checklists or templates for different analysis such as technical, economical or operational, and managing the rationale of feasibility analysis are included here.
 - **Attribute analysis.** User defined attributes that are assignable to each requirement such as risk, priority and cost provides metrics for tracking requirements based on project needs. The topics within this group of tool features are storing and managing attributes in various formats such as text, graphics or attachments, storing, sorting, grouping and ordering attributes, and managing changes to attributes.
 - **Risk analysis and management.** Risk analysis provides a standard approach with which to identify and document potential risks for the ongoing project, and risk management is an approach used to identify and document risk factors, evaluating their potential severity. Both approaches propose strategies for mitigating the identified risks. The tools might exchange information pertinent to risk analysis and management with external risk analysis and management tools, and maintain a relationship of requirements to risks raised and risks mitigated in order to perform risk analysis and management tasks.
- **Specification.** This category encompasses features which are focused on documenting the functions that a software or system must provide, and the constraints that it must respect. The requirements should be specified in a consistent, accessible and reviewable manner if this goal is to be accomplished.
 - **Requirements specification documentation.** The output from the entire requirements specification task should be stored, retrieved, and edited in various formats. Exporting the document in standard formats such as RTF or MS Word, generating presentation-quality charts and graphs in standard formats such as PNG or JPEG, and generating the output of the specification in a finished form are capabilities which might ease the requirements engineers' work.
- **Modeling.** This category includes features focused on the application of certain techniques to produce useful and verifiable requirements models.
 - **Modeling analysis.** Modeling analysis depicts the requirements at a high level of abstraction. These models include data-flow diagrams, entity relationship diagrams, or UML diagrams. The subjects tackled in this group of capabilities are importing and exporting to and from modeling tools, storing and managing templates for goal-oriented scenarios, and storing and managing user defined scenarios.
 - **Modeling and specification languages.** The storage and display of different modeling notations helps requirements engineers to represent requirements in an appropriate and useful manner, according to the complexity of the system. For more information, see Table 8.
- **Verification and validation (V&V).** This category includes capabilities aimed at supporting the various tests and means of evaluation involved in verifying and validating the requirements.
 - **V&V.** The RE tools might support these tasks by storing and managing verification/validation plans and verification/validation procedures, by supporting the review and inspection of such plans and procedures, and by providing standard format for interfacing verification/validation tools, among other features.
- **Management.** This category explores the tools' ability to support the monitoring of the changes and the maintenance of the requirements, thus ensuring that the requirements accurately reflect the product.
 - **Baseline of the requirements.** The baseline is the set of requirements which have been agreed and approved by the users and/or stakeholders. Some desirable features are storing and managing a baseline document, version controlling of baseline requirements, and read/write protection of baseline products.
 - **Requirements change management.** Change management tracks changes to requirements and ensures that approved changes are communicated to all affected stakeholders. This might be achieved by managing version identification (e.g. version number, date, time of creation or revision), providing check-in and out capabilities of the baseline to add, change and update requirements, and providing flexible search criteria for all requirements artefacts affected by the changed requirements.
 - **Project management.** Project management needs to keep track of the status of requirements and applies to managing resources, schedules, and commitment of them. The features included in this group are the recording, tracking, and reporting of the status of the overall requirements management process, and the exchange of information pertinent to project management with external project management tools.
 - **Open or closed data model.** An open data model is supported by ensuring that the fundamental data structures of the application have a standard format, thus allowing external components to access the information they need at runtime without requiring a complex protocol [42]. For more information, see Table 9.
- **Traceability.** This category includes capabilities focused on documenting the life of a requirement, providing linkage mechanisms between associated requirements, and tracking the changes made to each requirement.
 - **Traceability.** This group of tool features includes storing and managing the identification and documentation of stakeholders and their roles and responsibilities, displaying traceability in textual and/or graphical form, and tracing across tools boundaries.

- **Flexible tracing.** Different kind of traces might be defined between requirements. One to many and many to one, text to text, and text to graphics are some examples of tracing features included in this group.
- **Bi-directional tracing.** Bi-directional traceability of requirements helps to determine that all the source requirements have been completely addressed and that all lower level requirements can be traced to a valid source. Some useful features are the bi-directional tracing of requirements to system elements, the allocation of requirements to system elements, and user needs to requirements.
- **Traceability analysis.** Traceability analysis provides an insight into the overall project status. Generating a report for missing and/or orphan requirements, generating a report that identifies changes from/to a requirement throughout the development life cycle, and generating and exporting traceability matrix tables are some ways in which this might be achieved.
- **Other capabilities.** This category encompasses features related to the integration of the tool into the systems and software development environment.
 - **RE tool administrative information.** Since versions can be distributed on different hardware platforms in many projects, administrative information will be needed to monitor system-wide use. This information might be supported by providing user information indicating single user or multiple concurrent user environment, by providing platform information indicating which platforms and operating systems support the tool, and by providing database information indicating which database it uses (proprietary or commercial).
 - **Graphical user interface (GUI).** Various GUI capabilities should be supported to improve the usability of the tool. Providing standard Windows, UNIX, Mac environment capabilities (e.g. multi-windows, open multiple files concurrently), providing multiple windows or views in the tool (e.g. a change in one view will automatically reflect in all other views), and providing an interactive graphical input and manipulation of data are the capabilities included here.
 - **Data integration.** Different parts of the organization normally use different systems to produce, store, and search for their critical data, thus leading to the need to combine information from these various systems [43]. For more information, see Table 10.

References

- [1] T. Nakatani, H. Kaiya, Japanese workshop on requirements engineering tools (JWRET), in: Proc. of the 21st IEEE/ACM Int. Conf. on Autom. Softw. Eng., IEEE Computer Society, Washington, DC, USA, 2006, p. 378.
- [2] M. Hoffmann, N. Kühn, M. Weber, M. Bittner, Requirements for requirements management tools, in: Proc. of the 12th IEEE Int. Requir. Eng. Conf., IEEE Computer Society, Washington, DC, USA, 2004, pp. 301–308.
- [3] V.R. Basili, F. Shull, F. Lanubile, Building knowledge through families of experiments, IEEE Trans. Softw. Eng. 25 (1999) 456–473.
- [4] J.M. Carrillo de Gea, J. Nicolás, J.L. Fernández Alemán, A. Toval, C. Ebert, A. Vizcaíno, Requirements engineering tools, IEEE Softw. 28 (2011) 86–91.
- [5] A. Gregoriades, A. Sutcliffe, Scenario-based assessment of nonfunctional requirements, IEEE Trans. Softw. Eng. 31 (2005) 392–409.
- [6] R.J. Hall, A method and tools for large scale scenarios, Autom. Softw. Eng. 15 (2008) 113–148.
- [7] J. Kääriäinen, J. Koskela, P. Abrahamsson, J. Takalo, Improving requirements management in extreme programming with tool support – an improvement attempt that failed, in: Proc. of the 30th EUROMICRO Conf., IEEE Computer Society, Washington, DC, USA, 2004, pp. 342–351.
- [8] P. Giorgini, F. Massacci, J. Mylopoulos, N. Zannone, ST-Tool: a CASE tool for security requirements engineering, in: Proc. of the 13th IEEE Int. Requir. Eng. Conf., IEEE Computer Society, Los Alamitos, CA, USA, 2005, pp. 451–452.
- [9] N. Seyff, P. Grunbacher, N. Maiden, A. Tosar, Requirements engineering tools go mobile, in: Proc. of the 26th Int. Conf. on Softw. Eng., IEEE Computer Society, Washington, DC, USA, 2004, pp. 713–714.
- [10] N. Maiden, N. Seyff, P. Grunbacher, O. Tojare, K. Mitteregger, Making mobile requirements engineering tools usable and useful, in: Proc. of the 14th IEEE Int. Requir. Eng. Conf., IEEE Computer Society, Washington, DC, USA, 2006, pp. 26–35.
- [11] L. Jiang, A. Eberlein, A tool for requirements engineering process development, in: Proc. of the 31st Annual Int. Comput. Softw. and Appl. Conf., COMPSAC '07, IEEE Computer Society, Washington, DC, USA, 2007, pp. 319–325.
- [12] J. Portillo Rodríguez, C. Ebert, A. Vizcaíno, Technologies and tools for distributed teams, IEEE Softw. 27 (2010) 10–14.
- [13] F. Lanubile, Collaboration tools for global software engineering, C. Ebert, Collaboration tools for global software engineering, R. Prikładnicki, Collaboration tools for global software engineering, A. Vizcaíno, Collaboration tools for global software engineering, IEEE Softw. 27 (2010) 52–55.
- [14] K. Schmid, K. Krennrich, M. Eisenbarth, Requirements management for product lines: extending professional tools, in: Proc. of the 10th Int. Softw. Product Line Conf., IEEE Computer Society, Los Alamitos, CA, USA, 2006, pp. 113–122.
- [15] A. Toval, B. Moros, J. Nicolás, J. Lasheras, Eight key issues for an effective reuse-based requirements process, Comput. Sys. Sci. Eng. 23 (2008) 1–13.
- [16] O. Gotel, P. Mäder, How to select a requirements management tool: initial steps, in: Proc. of the 17th IEEE Int. Requir. Eng. Conf., IEEE Computer Society, Los Alamitos, CA, USA, 2009, pp. 365–367.
- [17] R.J. Hamann, M.J.A. Oort, Using a requirement management tool for verification management, Syst. Eng. 4 (2001) 169–178.
- [18] Y. Hong, M. Kim, S.-W. Lee, Requirements management tool with evolving traceability for heterogeneous artifacts in the entire life cycle, in: Proc. of the 8th ACIS Int. Conf. on Softw. Eng. Research, Management and Applications, SERA '10, IEEE Computer Society, Washington, DC, USA, 2010, pp. 248–255.
- [19] B. Johansson, R.A. de Carvalho, Software tools for requirements management in an ERP system context, in: Proc. of the 25th ACM Symp. on Appl. Computing, SAC '10, ACM, New York, NY, USA, 2010, pp. 169–170.
- [20] M. Heindl, F. Reinisch, S. Biffl, A. Egyed, Value-based selection of requirements engineering tool support, in: Proc. of the 32nd EUROMICRO Conf. on Softw. Eng. and Adv. Appl., IEEE Computer Society, Washington, DC, USA, 2006, pp. 266–273.
- [21] B. Alenljung, A. Persson, DESCRY: a method for evaluating decision-supporting capabilities of requirements engineering tools, in: Proc. of the 14th Int. Conf. on Requir. Eng.: Found. for Softw. Qual., REFSQ '08, Springer, 2008, pp. 52–57.
- [22] R. Schmid, J. Ryser, S. Berner, M. Glinz, R. Reutemann, E. Fahr, A survey of simulation tools for requirements engineering, Technical Report, German Informatics Society, 2000.
- [23] D. Zowghi, C. Coulin, Requirements elicitation: a survey of techniques, approaches, and tools, in: A. Aurum, C. Wohlin (Eds.), Engineering and Managing Software Requirements, Springer, 2005, pp. 19–46.
- [24] D. Beuche, A. Birk, H. Dreier, A. Fleischmann, H. Galle, G. Heller, D. Janzen, I. John, R.T. Kolagari, T. von der Massen, A. Wolfram, Using requirements management tools in software product line engineering: the state of the practice, in: Proc. of the 11th Int. Softw. Product Line Conf., IEEE Computer Society, Washington, DC, USA, 2007, pp. 84–96.
- [25] D. Mellado, E. Fernández-Medina, M. Piattini, A common criteria based security requirements engineering process for the development of secure information systems, Comput. Stand. Interfaces 29 (2007) 244–253.
- [26] ISO/IEC JTC 1 SC 7, ISO/IEC TR 24766:2009, Information Technology – Systems and Software Engineering – Guide for Requirements Engineering tool Capabilities, ISO, Geneva, Switzerland, first ed., 2009.
- [27] ISO/IEC JTC 1 SC 7, ISO/IEC 14102:2008, Information Technology – Guideline for the Evaluation and Selection of CASE Tools, ISO, Geneva, Switzerland, second ed., 2008.
- [28] V. Basili, H. Rombach, The TAME project: towards improvement-oriented software environments, IEEE Trans. Softw. Eng. 14 (1988) 758–773.
- [29] C. Wohlin, P. Runeson, M. Höst, M.C. Ohlsson, B. Regnell, A. Wesslén, Experimentation in Software Engineering: An Introduction, Kluwer Academic Publishers, 2000.
- [30] M. Staron, W. Meding, C. Nilsson, A framework for developing measurement systems and its industrial evaluation, Inf. Softw. Technol. 51 (2009) 721–737.
- [31] B. Kitchenham, DESMET: a method for evaluating software engineering methods and tools, Tech. Report TR96-09, Dept. of Computer Science, University of Keele, 1996.
- [32] J. Dougherty, R. Kohavi, M. Sahami, Supervised and unsupervised discretization of continuous features, in: Machine Learning: Proc. of the 20th Int. Conf., Morgan Kaufmann, 1995, pp. 194–202.
- [33] L. Ott, M. Longnecker, An Introduction to Statistical Methods and Data Analysis, Brooks/Cole, Cengage Learning, sixth ed., 2010, pp. 56–139.
- [34] J. Cohen, A coefficient of agreement for nominal scales, Educat. Psychol. Measur. 20 (1960) 37–46.
- [35] J. Cohen, Weighted kappa: nominal scale agreement provision for scaled disagreement or partial credit, Psychol. Bull. 70 (1968) 213–220.
- [36] J.R. Landis, G.G. Koch, The measurement of observer agreement for categorical data, Biometrics 33 (1977) 159–174.
- [37] D. Altman, Practical Statistics for Medical Research, Texts in Statistical Science Series, Chapman and Hall, 1991.

- [38] J. Fleiss, B. Levin, M. Paik, *Statistical Methods for Rates and Proportions*, third ed., J. Wiley, 2003.
- [39] L. Bratthall, M. Jørgensen, Can you trust a single data source exploratory software engineering case study?, *Emp Softw. Eng.* 7 (2002) 9–26.
- [40] S. Draper, *The Hawthorne, Pygmalion, Placebo and other effects of expectation: some notes*, 2011.
- [41] T.C. Lethbridge, S.E. Sim, J. Singer, Studying software engineers: data collection techniques for software field studies, *Emp. Softw. Eng.* 10 (2005) 311–341.
- [42] B. Myers, *The case for an open data model*, Technical Report, Carnegie Mellon University, 1998.
- [43] L.M. Haas, E.T. Lin, M.A. Roth, Data integration through database federation, *IBM Syst. J.* 41 (2002) 578–596.
- [44] OMG, *Requirements Interchange Format (ReqIF)*, v1.0.1, OMG, 2011.
- [45] M.R. Monteiro, C. Ebert, M. Recknagel, Improving the exchange of requirements and specifications between business partners, in: *Proc. of the 17th IEEE Int. Requir. Eng. Conf.*, IEEE Computer Society, Washington, DC, USA, 2009, pp. 253–260.